What is claimed is:

1. A system comprising:

a first port to receive a network packet;

a second port in communication with the first port, the second port to transmit the network packet after processing;

circuitry to associate first control information with a first portion of the network packet and to associate second control information with a second portion of the network packet;

circuitry to process the first portion of the network packet and to process the second portion of the network packet at least partially in parallel with processing the first portion of the network packet; and

circuitry to enqueue the first portion and the second portion for transmission to a second port in the same order in which the first portion and the second portion were received at the first port.

2. The system of claim 1 wherein the circuitry comprises:

one or more peripheral buses;

a memory system;

a processor coupled to the one or more peripheral buses and the memory system, the processor adapted to forward data from the first port to the second port; and

a bus interface to receive the first portion of the network packet and the second portion of the network packet from the first port and enqueueing the first portion and the second portion in the order in which they were received from the first port for transmission to the second port, the first and second portions being processed at least partially in parallel.

3. The system of claim 2 wherein the processor comprises one or more microengines to execute program threads, the threads include receive schedule program threads to assign the first portion of the network packet from the first port to a first receive processing program thread and the second portion of the network packet to a second receive processing program thread, wherein the bus interface is responsive to the one or more microengines, and wherein the first and second receive processing program threads are adapted for processing and enqueueing.

4. The system of claim 3 wherein the bus interface uses sequence numbers to enqueue the first portion and the second

portion, wherein the bus interface is to associate one or more first portion sequence numbers with the first portion and one or more second portion sequence numbers with the second portion as the first and second portions are received from the first port.

5. The system of claim 4 wherein the bus interface is further to maintain a second set of sequence numbers for use by the first and second receive processing program threads in determining the order in which the first and second portions are to be enqueued.

6. The system of claim 2 wherein the one or more peripheral buses comprise at least one input-output bus, wherein the processor is adapted to interface over the input-output bus with at least one of a media access controller device and a high-speed device, the high-speed device comprising at least one of a gigabit Ethernet MAC and a dual gigabit MAC with two ports.

7. The system of claim 2 wherein the memory system further comprises at least one of a random access memory, a synchronous dynamic random access memory, a synchronous

dynamic random access memory controller, a static random access memory controller, and a nonvolatile memory.

8. The system of claim 7 wherein the memory system further comprises a memory bus, wherein the memory bus is adapted to couple one or more bus interfaces to one or more memory controllers.

9. The system of claim 2 wherein the processor comprises one or more microengines to execute program threads, wherein the one or more microengines are configured to operate with shared resources, and wherein the shared resources comprise the memory system and the one or more peripheral buses.

10. The system of claim 9 wherein the bus interface comprises an input-output bus interface.

11. The system of claim 9 wherein the bus interface is coupled to an input-output bus, wherein the input-output bus is coupled to a dual gigabit MAC.

12. The system of claim 9 wherein at least one of the microengines comprises:

a control store for storing a microprogram; and

a set of control logic, wherein the set of control logic comprises an instruction decoder and one or more program counter units.

13. The system of claim 12 wherein at least one of the microengines further comprises a set of context event switching logic to receive messages from the shared resources.

14. A communication system comprising:

a media access controller capable of providing one or more status flags, the media access controller comprising one or more ports;

a bus interface unit comprising one or more registers, wherein the one or more registers comprise control registers and status registers;

a bus connected between the media access controller and the bus interface unit; and

a sequencer to poll the one or more status flags and place the one or more status flags to the one or more registers over the bus, wherein the communication system is capable of processing one or more packets of data, and wherein the communication system is capable of maintaining an intra-

packet order and an inter-packet order for the one or more ports.

15. The communication system of claim 13 wherein the media access controller further comprises one or more transmit registers and one or more receive registers, and wherein the one or more ports comprise at least two gigabit Ethernet ports.

16. The communication system of claim 15 wherein the communication system is capable of enqueuing a first portion of a network packet and a second portion of a network packet for transmission to a second port in the same order in which the first portion and the second portion were received at a first port.

17. The communication system of claim 16 wherein the one or more status flags comprise one or more transmit status flags and one or more receive status flags, and wherein the one or more flags indicate whether an amount of data in associated transmit registers and associated received registers have reached a threshold level.

18. The communication system of claim 17 wherein a receive scheduler thread uses the one or more registers in the bus interface unit to determine how to issue a receive request.

19. The communication system of claim 16 wherein the communication system uses a set of sequence numbers for each port, wherein the sequence numbers comprise a network packet sequence number, a MAC packet sequence number, and an enqueue sequence number.

20. A method of supporting one or more network port modes comprising:

determining an assignment of one or more receive processing threads to process packet data in a queue, wherein the one or more network port modes comprise a single thread mode, a header/body thread mode, and an explicit thread mode;

determining an availability of one or more threads to process a received request;

determining an execution time for the one or more received threads; and

determining an end of a network packet and a beginning of a successive network packet.

21. The method of claim 20 wherein the single thread mode assigns a single thread to a packet for one or more speculative requests.

22. The method of claim 20 wherein the header/body mode assigns two threads to process a MKPT within the network packet, wherein the first thread comprises the header thread and the second thread is the body thread, and wherein the header thread is used to determine how to forward the network packet and the body thread is used for moving a remainder of the network packet to memory.

23. The method of claim 20 wherein the explicit thread mode comprises providing one or more received processing threads with an identification of a thread assigned to the next MPKT receive request, wherein each assigned thread provides the next assigned thread with a pointer to a memory.

24. A system comprising:

receiving means to receive a network packet at a first port;

transmitting means for transmitting the network packet after processing, the receiving means in communication with the transmitting means;

means for associating a first control information with a first portion of the network packet;

means for associating second control information with a second portion of the network packet;

means for processing the first portion of the network packet and the second portion of the network packet at least partially in parallel; and

means for enqueueing the first portion and the second portion for transmitting to a second port in the same order in which the first portion and the second portion were received at the first port.

25. The system of claim 24 wherein the means for processing the first portion of the network packet and the second portion of the network packet at least partially in parallel is implemented at least partially in software.